

**Core-i RST - REST/Webservice Framework  
Core-i ENC - DB2 Data Encryption  
Load Testing Results**

**Summary**

**#Tables Accessed By API:** 2  
**#Rows In Each Table:** 1,000,000  
**Test Location:** USA  
**Endpoint Location:** Germany  
**Encrypted Column:** Account# - zoned(10,0)

Apache Jmeter is used to conduct load tests over NON-ENCRYPTED account column utilizing the CoreiRST framework. Those stats are recorded for both http and https at different load test amounts.

A second round of testing is executed using the same tests, however this time CoreiENC has implemented DB2 at rest data encryption on the account column for all 1M records. Additionally, the user that is used in the http request is set to have encrypted values masked, therefore the account number is returned to the off platform consumer (Apache Jmeter) with the masked account number value. (FIRST\_6 = 9's)

So one round trip of an http request/response with CoreiENC in play looks like:

- 1.) Apache Jmeter executes the http request
- 2.) The request reaches the IBMi endpoint sitting in Germany.
- 3.) The IBMi sqlrpgle program which is used in the CoreiRST framework is invoked to build a json response.
- 4.) The sql/json function retrieves a customer number from the testcust and testaccount table where testaccount.aacct column (bank account number) is encrypted.
- 5.) The field procedure program attached to that column fires and decrypts the value, but returns the masked value since the user being used in the http request has masking set by CoreiENC.
- 6.) The masked value is retrieved by the sql/json query and is returned back to the http request ultimately received back to Apach Jmeter.

Retrieved http/json response structure with masked account#...

```
{
  "success": 1,
  "resultMessage": "Success",
  "list": [
    {
      "custNo": "11111",
      "custInfo": {
        "firstName": "First Name: 11111",
        "lastName": "Last Name: 11111",
        "address1": "Address1: 11111",
        "address2": "Address2: 11111",
        "city": "City: 11111",
        "state": "11",
        "zip": "11111",
        "routing": "11111",
        "accountNo": "9999991111"
      }
    }
  ]
}
```

Overall, CoreiENC impacted the performance of the CoreiRST load test with a **%10.5** latency increase.

## Detail

(150 users - 10sec ramp up)

----- w/ CoreiENC (encrypted/masked) -----

Protocol	Auth	Latency Avg/ms	Deviation	IBMi API Avg/ms	Latency Avg/ms	Deviation	IBMi API Avg/ms
HTTP	user/pass	281	13	.004624	311	16	.021203
HTTPS		594	75	.006625	676	323	.030090

Performance Hit Due To Encryption: %12.8 Latency Increase

(500 users - 20sec ramp up - 2 loops)

Protocol	Auth	Latency Avg/ms	Deviation	IBMi API Avg/ms	Latency Avg/ms	Deviation	IBMi API Avg/ms
HTTP	user/pass	291	135	.028280	300	147	.033929
HTTPS		816	496	.081681	830	321	.093691

Performance Hit Due To Encryption: %2.1 Latency Increase

(1000 users - 40sec ramp up - 2 loops)

Protocol	Auth	Latency Avg/ms	Deviation	IBMi API Avg/ms	Latency Avg/ms	Deviation	IBMi API Avg/ms
HTTP	user/pass	334	210	.023193	392	166	.040987
HTTPS		1011	578	.036158	1177	631	.108676

Performance Hit Due To Encryption: %16.7 Latency Increase

NOTE: best of 2 separate runs is recorded